Application No.09/843,609
After Non-Final Office Action of March 9, 2006

Claims

1.   (Currently Amended) A digital data computing method comprising:

utilizing a set of secured instructions and secured memory local to a client to execute, on the client, a process that makes requests and that requires at least asynchronous responses to those requests in order to continue operation, wherein at least one of said requests is associated with at least one of allocation and de-allocation of dynamic memory;

generating on a server, those responses external to the process and supplying them to that process, wherein a said response to a said request for at least one of allocation and/or de-allocation of dynamic memory includes data necessary to perform a respective one of at least said allocation and/or de-allocation of memory;

the an executing step including continuing operation of the process when at least asynchronous responses are received to the requests, and otherwise discontinuing operation of the process, wherein the executing step uses said data for at least one of allocation and/or de-allocation of dynamic memory associated with the process;

there being no real-time dependency of the process on to those responses, responses while execution operation of the process is continuing.

2,   (Cancelled)

3.   (Previously Presented) The method of claim 1, comprising performing the executing step on a server that comprises a secured coprocessor local to the client.

4.   (Original) The method of claim 1, comprising performing the executing step on a server that is remote with respect to the client.

5.   (Cancelled)

6.   (Original) The method of claim 1, wherein it is computationally difficult to unauthorizedly simulate generation of the responses.

2

Application No.09/843,609   Docket No.: 102788-0009
After Non-Final Office Action of March 9, 2006

7.   (Original) The method of claim 6, wherein the executing step includes executing
        transformed code and wherein it is computationally difficult to determine proper
        responses to the requests without access to at least a portion of that code prior to a
        transformation that produces that transformed code.

8.   (Original) The method of claim 7, comprising performing the transformation
        automatically.

9.   (Original) The method of claim 7, comprising performing the transformation manually.

10.   (Currently Amended) A digital data computing method comprising:

        utilizing a set of secured instructions and secured memory local to a client to execute, on
        the client, a process that makes requests and that requires at least asynchronous responses
        to those requests, wherein at least one of said requests is associated with at least one of
        allocation and de-allocation of dynamic memory;

        generating on a server, those responses external to the process and supplying them to that
        process wherein a said response to a said request for at least one of allocation and/or de-
        allocation of dynamic memory includes data necessary to perform a respective one of at
        least said allocation and/or de-allocation of memory;

        the an executing step including continuing execution of the process when at least
        asynchronous responses are received to the requests, and otherwise discontinuing
        execution of the process, wherein a said executing step uses said data for at least one of
        allocation and/or de-allocation of dynamic memory associated with the process, there
        being no real-time dependency of the process on to those responses while execution
        operation of the process is continuing; and

        wherein the generating step includes generating non-deterministic responses to the
        requests.

11.   (Currently Amended) A digital data computing method comprising:

3

utilizing a set of secured instructions and secured memory local to a client to execute, on the client, a process that makes requests and that requires at least asynchronous responses to those requests, wherein at least one of said requests is associated with at least one of allocation and de-allocation of dynamic memory;

generating on a server, those responses external to the process and supplying them to that process, wherein a said response to a said request for at least one of allocation and/or de-allocation of dynamic memory includes data necessary to perform a respective one of at least said allocation and/or de-allocation of memory

the an executing step including continuing execution of the process when at least asynchronous responses are received to the requests, and otherwise discontinuing execution of the process, wherein said executing step uses said data for at least one of allocation and/or de-allocation of dynamic memory associated with the process, there being no real-time dependency of the process on to those responses, responses while execution operation of the process is continuing; and

wherein the executing step includes executing transformed code and wherein it is computationally difficult to generate the non-deterministic response without access to at least a portion of that code prior to a transformation that produces that transformed code.

12.   (Original) The method of claim 1, wherein the executing and generating steps are adapted to securing the generation of responses against any of unauthorized use, access, copying and functional analysis, and of controlling the execution of the process.

13.   (Currently Amended) A digital data computing method securing and controlling a set of instructions (hereafter, "code") against at least one of unauthorized use, access, copying and functional analysis comprising:

including within the code on a client a set of secured instructions and secured memory local to the client, where the instructions and memory are secured either by hardware or software, requests to which the code requires at least asynchronous responses in order to continue operation, wherein at least one of said requests is associated with at least one of allocation and de-allocation of dynamic memory;

4

generating, on a server, ~~these~~ responses external to the code and supplying them to the client, <u>wherein a said response to a said request for at least one of allocation and/or de-allocation of dynamic memory includes data necessary to perform a respective one of at least said allocation and/or de-allocation of memory;</u>

continuing, on the client, operation of the code when at least asynchronous responses are received to the requests, and otherwise discontinuing the operation of the code, <u>wherein said executing step uses said data for at least one of allocation and/or de-allocation of dynamic memory associated with the process,</u> ~~there~~-being no real-time dependency of the code <u>on</u> ~~to those~~ responses, while operation of the code continuing.

14   (Original) The method of claim 13, wherein the code is comprised of high-level language or object code or any intermediary level set of computer instructions, or microcode.

15.   (Original) The method of claim 13, including the step of performing a transformation that includes generating any of code and data upon which the responses are based.

16.   (Original) The method of claim 15, comprising performing the transformation automatically.

17.   (Original) The method of claim 15, wherein performing the transformation manually.

18.   (Original) The method of claim 13, wherein it is computationally difficult to unauthorizedly simulate the generation of proper responses to the requests.

19.   (Original) The method of claim 18, wherein it is computationally difficult to generate the proper responses without access to at least a portion of code prior to the transformation.

20.   (Original) The method of claim 13, wherein the generating step includes a non-deterministic action.

21.   (Original) The method of claim 20, wherein it is computationally difficult to determine the effect of the non-deterministic action without access to at least a portion of the code prior to a transformation that produces that transformed code.

5

22.   (Original) The method of claim 15, comprising performing executing the code subsequent to transformation on the client.

23.   (Original) The method of claim 22, comprising performing the executing step on a server that comprises secured coprocessor local to the client.

24.   (Original) The method of claim 22, wherein the server is a remote processor.

Claims 25-43 have been cancelled.

6